ORDERING THE t LARGEST OF n ITEMS USING BINARY COMPARISONS*

by

Abdollah Hadian and Milton Sobel

Technical Report No. 136

February 1970

*This research was supported by the National Science Foundation under Grant No. NSF-GP 11021.

1. Introduction.

There are given n numbers, or items with scalar attributes, $x_1, x_2, \ldots, x_n$ which are unknown and assumed to be pairwise unequal. For given $t(1 \leq t \leq n-1)$ we wish to find and order the t largest of these items using only binary errorless comparisons; each comparison between two x's tells us only which is larger and which is smaller.

As in other similar problems [4], [9], and [3], one can consider two optimality criteria for evaluating and comparing procedures that achieve the prescribed goal. A minimax (M-optimal) procedure minimizes the maximum number of comparisons needed. Assuming a random ordering at the outset (with equal probabilities for each of the $n!$ arrangements), an E-optimal procedure minimizes the expected number of comparisons required.

The M-optimal procedure for $t = 2$ has been considered by J. Schreier [7] and J. Slupecki [8] (see also Steinhaus [10]). Several procedures are given in [9] with emphasis on information-theoretic methods; two of these procedures are shown to be M-optimal. The minimax value and the expectations for these procedures are given in a table for $n = 2(1)10$.

The M-optimal for $t = n - 1$ (ordering all the items) was originally considered by Steinhaus in [10], [11], and [12]. In [10] a procedure was given which was first conjectured to be M-optimal and later shown by Steinhaus not to be M-optimal. However it was shown by Hadian [4] that in a restricted class of procedures the set of procedures that are both M-optimal and E-optimal in this class includes the original procedure of Steinhaus. (For further discussion of the special case $t = n - 1$ see also [4] and the introduction of [9].) Ford and Johnson [2] also give a procedure for $t = n - 1$ which is known to be M-optimal for certain small values of n.

In this paper we give two procedures R and R' for finding and ordering the t largest of n items and find the maximum number of comparisons $M(R_{t,n}) \equiv M_t(n)$ required under procedures R; the value of $M(R'_{t,n}) \equiv M'_t(n)$ is conjectured to be equal to $M_t(n)$ for all t and n. The procedures R and R' are shown to be M-optimal for $t = 1$ and 2. For complete ordering $M_{n-1}(n) = M_{n-1}(n|R_S)$, where R_S is the Steinhaus procedure [10]. The latter is an inductive procedure and assuming that $j - 1$ items are ordered we compare the $j^{\underline{th}}$ item with the middle one (or one of the two middle ones) of $j - 1$ items, then with the middle one of the left or right subset, etc., until the $j^{\underline{th}}$ item is inserted and we have j ordered items $(j = 2, 3, \ldots, n)$.

Our problem is related to the problem of finding the $t^{\underline{th}}$ largest of n items but differs from it by the fact that in our problem one finds the $j^{\underline{th}}$ largest for $j = 1, 2, \ldots, t$ and orders them. The $t^{\underline{th}}$ largest problem was considered by Kislicyn [5] and by Hadian and Sobel [3] (see also page 48 of [6]). It is of interest to note that the maximum number of comparisons $P_t(n)$ obtained by Kislicyn for finding the $t^{\underline{th}}$ largest is equal to $M_t(n)$ obtained below for finding the t largest. This suggests that Kislicyn's result in [5] for finding the $t^{\underline{th}}$ largest could be improved; such an improvement was actually accomplished in [4].

2. The procedure $R_{1,n}$.

The procedure $R_{t,n}$ depends heavily on $R_{1,n}$ (finding the best one of n) and we therefore describe $R_{1,n}$ first in some detail. Although there are many ways of finding the best one of n items and they all require exactly $n - 1$ comparisons (without any variation), we are interested in a particular method that leads to a desirable recursion.

The procedure $R_{1,n}$ is obvious for $n = 2$ items, since only 1 step is required. For $n > 2$ we partition n by writing $n = [\frac{n}{2}] + \{\frac{n}{2}\}$, where $[\frac{n}{2}]$ is the integer part of $\frac{n}{2}$ and $\{\frac{n}{2}\} = n - [\frac{n}{2}]$. Procedure $R_{1,n}$ is then defined by "semi-induction," i.e., we find the largest in each of the two halves using $R_{1,[\frac{n}{2}]}$ and $R_{1,\{\frac{n}{2}\}}$, and then use one more comparison between the largest items in each of these two subsets.

Procedure $R_{1,n}$ has the following properties; an additional property will be given later.

1. $M_1(n) = n - 1$.

2. The maximum number of "seconds," i.e., items eligible to be second largest among all n items is $\{\log n\}$.

Here $\{x\}$ is the smallest integer equal to or greater than x and all logs are to base 2 unless stated otherwise.

To prove property (1) we notice that $M_1(n)$ satisfies the following recursive formula for $n \geq 2$

$$(2.1) \qquad M_1(n) = M_1(\{\tfrac{n}{2}\}) + M_1([\tfrac{n}{2}]) + 1,$$

where $M_1(1) = 0$. This follows because the largest of the subsets of sizes $\{\frac{n}{2}\}$ and $[\frac{n}{2}]$ are found by using $R_{1,\{\frac{n}{2}\}}$ and $R_{1,[\frac{n}{2}]}$, respectively. The number 1 on the RHS of (2.1) stands for the one comparison needed to find the larger of the two largest items in the subsets of size $\{\frac{n}{2}\}$ and $[\frac{n}{2}]$. Clearly $M_1(2) = 1$ and, since $[\frac{n}{2}] \leq \{\frac{n}{2}\} < n$ for $n > 2$ and by induction $M_1(k) = k - 1$ for $k < n$, we obtain from (2.1)

$$(2.2) \qquad M_1(n) = (\{\tfrac{n}{2}\} - 1) + ([\tfrac{n}{2}] - 1) + 1 = n - 1.$$

To prove property 2, let $S(n)$ denote the maximum number of seconds after completing the procedure $R_{1,n}$. By the definition of procedure $R_{1,n}$, $S(n)$ satisfies the recursion

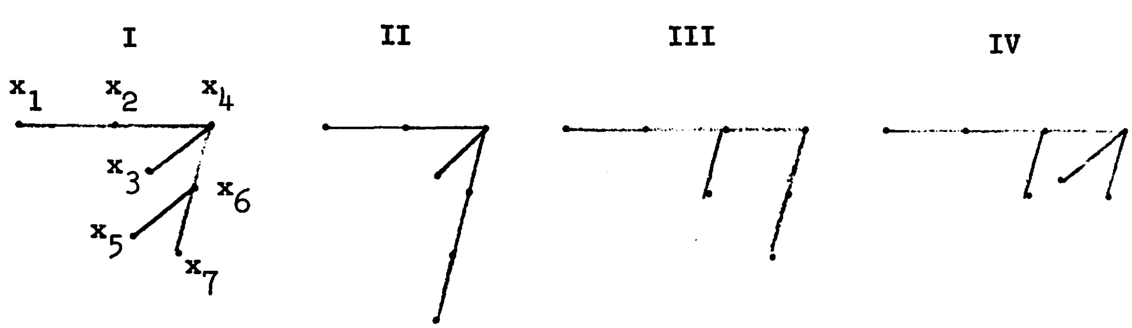
$$(2.3) \qquad S(n) = S(\{\tfrac{n}{2}\}) + 1 \;;$$

this maximum is attained if in the last comparison the maximum comes from the subset of size $\{\frac{n}{2}\}$. Clearly $S(2) = 1$ and for $n > 2$, since $\{\frac{n}{2}\} < n$ and by induction $S(k) = \{\log k\}$ for $k < n$, we obtain from (2.3)

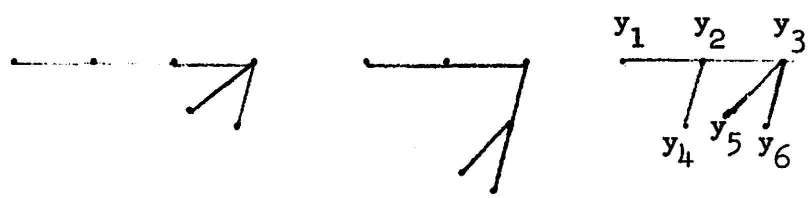
$$(2.4) \qquad S(n) = \{\log\{\tfrac{n}{2}\}\} + 1 = \{\log \tfrac{n}{2}\} + 1 = \{\log n\}.$$

Another property of $R_{1,n}$ is the so-called recursive property which enables us to remove the largest item and rebuild another "hierarchical" structure which is similar to that obtained by $R_{1,n-1}$ if we started with $n - 1$ items. When we remove the largest of n items, we are left with at most $\{\log n\}$ connected subsets, which we consider in order of their size. Comparing the largest item in the smallest subset with the largest in the second smallest, the resulting maximum with the largest of the third smallest, etc., we build up a connected structure of size $n - 1$ for finding the second largest of all n items, This "patching" procedure which takes at most $\{\log n\} - 1$ comparisons follows one of the patterns that arises in $R_{1,n-1}$ and, in particular, the maximum number of seconds in the final connected structure of size $n - 1$ is at most $\{\log (n-1)\}$. For example, if $n = 7$ then $R_{1,7}$ results in one of the four structures:

(2.5)



where items x_i, x_j with x_i to the left of x_j) are ordered $(x_i < x_j)$ if a line segment connects them. Thus in I we have $x_1 < x_2 < x_4$, $x_3 < x_4$, $x_5 < x_6 < x_4$ and $x_7 < x_6$. Removing x_4 (and the line segments connected to x_4), we are left with 3 connected subsets of sizes 1, 2 and 3. Then it takes 2 comparisons, x_3 vs. x_2 followed by the $\max(x_3, x_2)$ vs. x_6, to find the second best of the 7 items. This results in one of three possible structures:

(2.6) 

each of which has at most $3 = \{\log 6\}$ seconds (i.e., thirds for the original problem with 7 items). These are among the same structures that we would have obtained if we started with 6 items and used procedure $R_{1,6}$. We note that a repetition of removing the largest item in the last structure in (2.6) leads to 2 more comparisons y_5 vs. y_6 and $\max(y_5, y_6)$ vs. y_2 and the maximum number of seconds (i.e., fourths for the original problem with 7 items) will be $3 = \{\log 5\}$, etc.

It follows from the above discussion that if we want to find and order the t best of 7 items $(1 \leq t \leq 6)$, then the maximum number of comparisons will be $M_1(7) = 6$ and for $2 \leq t \leq 6$

$$M_t(7) = 7 - t + \sum_{j=9-t}^{7} \{\log j\} . \tag{2.7}$$

A general proof of this result for any n and t is given below.

3. <u>The procedure $R_{t,n}$ for $t > 1$.</u>

1. We start by finding the largest of n items using procedure $R_{1,n}$. Removing the largest item (and the line segments connected to it), either the subset of size $[\frac{n}{2}]$ or of size $\{\frac{n}{2}\}$ remains unaltered. Let m denote the size of the subset that is broken up so that only $m - 1$ items remain in it.

2. We make use of all the comparisons that still remain in the connected pieces comprising the subset of size $m - 1$ and continue to find the largest of the $m - 1$ items using procedure $R_{1,m-1}$. This amounts to arranging the connected subsets in order of size, comparing the largest item in the smallest subset with the largest item in the second smallest subset, the resulting maximum with the largest item in the third smallest subset, etc., until the largest of $m - 1$ items is found. (Remark 1 of Section 6 deals with connected subsets of equal size.)

3. Compare the largest of the two subsets of size $m - 1$ and $n - m$ to find the largest of the remaining $n - 1$ items.

4. Remove the largest of $n - 1$ items and continue as in steps 2 and 3 above until the t largest of the original n items are found.

4. <u>Derivation of $M_t(n)$.</u>

The first step above requires $n - 1$ comparisons. Since we are interested in the maximum number of comparisons we take m to be $\{\frac{n}{2}\}$ and, using (2.4) with n replaced by $\{\frac{n}{2}\}$, the number of connected subsets or seconds from the subset of size $\{\frac{n}{2}\}$ is

$$\{\log\{\tfrac{n}{2}\}\} = \{\log \tfrac{n}{2}\} = \{\log n\} - 1$$

and hence the number of comparisons in step 2 above is $\{\log n\} - 2$.

Steps 3 and 4 are included as part of our induction and would require $M_{t-1}(n-1)$ comparisons, except that in the two connected subsets of size $[\frac{n}{2}]$ and $\{\frac{n}{2}\} - 1$ we already have $[\frac{n}{2}] - 1 + \{\frac{n}{2}\} - 2 = n - 3$ comparisons from steps 1 and 2 that we can use. Hence we obtain

$$(4.1) \qquad M_t(n) = n - 1 + \{\log n\} - 2 + M_{t-1}(n-1) - (n-3)$$

or

$$(4.2) \qquad M_t(n) - M_{t-1}(n-1) = \{\log n\}.$$

By simple iteration on t, we easily obtain

$$(4.3) \qquad M_t(n) = n - t + \sum_{j=n-t+2}^{n} \{\log j\}.$$

Remarks: This is equivalent to the result of Kislicyn [5]

$$(4.4) \qquad P_t(n) = n - 1 + \sum_{i=1}^{t-1} [\log(n-i)]$$

for the problem of finding the $t^{\underline{th}}$ largest of n items, which should require fewer comparisons than the problem under discussion. For $t = n - 1$ this formula coincides with the result of Steinhaus for ordering all n items, namely,

$$(4.5) \qquad M_{n-1}(n|R_S) = \sum_{j=2}^{n} \{\log j\}.$$

For $t = 2$, the equation (4.3) gives the result

$$(4.6) \qquad M_2(n) = n - 2 + \{\log n\} = n - 1 + [\log(n-1)],$$

which was shown to be M-optimal in [9]. We note also from (4.3) that $M_n(n) = M_{n-1}(n)$, as it should be.

Another expression for the result (4.3) without any summation is obtained by using the fact that for any integer $k \geq 2$

$$\sum_{j=2}^{k} \{\log j\} = k\{\log k\} - 2^{\{\log k\}} + 1 \; ; \tag{4.7}$$

the proof of (4.7) is straightforward and therefore omitted. Using (4.7), we obtain from (4.3)

$$\begin{aligned} M_t(n) &= n - t + \sum_{j=2}^{n} \{\log j\} - \sum_{j=2}^{n-t+1} \{\log j\} \\ &= n - t + n\{\log n\} - 2^{\{\log n\}} - (n-t+1)\{\log(n-t+1)\} + 2^{\{\log n-t+1\}} . \end{aligned} \tag{4.8}$$

For the particular subsequence of interest in [3] given by

$$n = 2^{r+1} - 3, \; t = 2^r - 1 \tag{4.9}$$

we obtain

$$M_t(n) = r(2^r - 2) + 2^{r+1} - 5. \tag{4.10}$$

This particular sequence was found in [3] to be "adverse" for the formula (4.3).

5. An alternative procedure $R'_{t,n}$.

This procedure depends on the concept of complete pairing defined for any n and on the idea of a knock-out tournament (see, e.g., David [1]) for the special case when we have $n = 2^r$ items. For any n, let the binary expansion of n be

$$n = 2^{r_1} + 2^{r_2} + \ldots + 2^{r_s} \qquad (r_1 > r_2 > \ldots > r_s \geq 0) \tag{5.1}$$

so that s is the number of ones in the binary notation for n. We partition our n items at random into s subsets according to (5.1) and perform a knock-out tournament for each subset of size 2^{r_j} $(j = 1,2,\ldots,s)$.

The Procedure $R'_{t,n}$.

1. First we do complete pairing on the n items and the s connected subsets are arranged in order of size. Let the largest items then be denoted by $x^{(1)}, x^{(2)},\ldots, x^{(s)}$.

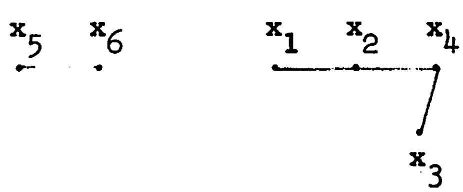
2. We then make further comparisons among these $x^{(j)}$ as follows: $x^{(1)}$ vs. $x^{(2)}$, $\max(x^{(1)}, x^{(2)})$ vs. $x^{(3)}$, etc., until the largest of these s items (which is also the largest of all n items) is found.

3. The largest item is removed (together with all line segments connected to it) and the remaining connected subsets are again arranged in order of size. As in step 2 we compare the largest items, starting from the smallest connected subset, to find the largest of the remaining $n - 1$ items. (Remark 1 of Section 6 deals with connected subsets of equal size.)

4. Repeat step 3 until the t largest are found; they are automatically ordered.

For example suppose $n = 6 = 2^2 + 2$ so that $s = 2$ and complete pairing in step 1 gives rise to the structures

(5.2)



For step 2 we compare x_6 vs. x_4 and obtain either of the two structures

(5.3)

In step 3 we remove the largest item (e.g., x_4 in the left figure above and are left with the three connected subsets (• •—• •—•). In step 4 we repeat this process until that t largest items are obtained.

We conjecture that $M'_t(n)$ is exactly the same as $M_t(n)$ given in (4.3). It is shown by calculations in Tables I and II that the procedures $R_{t,n}$ and $R'_{t,n}$ in general have a different expectation. In fact we note that the expectation under $R_{t,n}$ is never larger than that under $R'_{t,n}$ for any of the values of t and n used in Tables I and II (and it appears to continue to hold for all t and n).

6. <u>Remarks on procedures $R_{t,n}$ and $R'_{t,n}$</u>.

1. In step 2 of the procedure $R_{t,n}$ and step 3 of $R'_{t,n}$ we may get connected subsets of equal sizes. In this case we consider a connected subset to be 'smaller' if the number of permutations of items in that subset (consistent with the structure of the subset) is smaller. For example, the subset of size 4 with structure  consists of 2 permutations whereas the subset of the same size with structure  has 3 permutations.

2. Procedures $R_{t,n}$ and $R'_{t,n}$ are generalizations of procedures R_{I^*} and R_M, respectively, considered in [9] for $t = 2$; where they were shown to be M-optimal. Let $A_t(n) \equiv E\{R_{t,n}\}$ and $A'_t(n) \equiv E\{R'_{t,n}\}$ denote the expected number of comparisons required for finding and ordering the t largest of n items under procedures $R_{t,n}$ and $R'_{t,n}$, respectively. It is shown in [9, equations (3.17) and (3.9)] that

$$(6.1) \qquad A_2(n) = n + [\log n] - \frac{2^{1+[\log n]}}{n}$$

and

$$(6.2) \qquad A_2'(n) = n - 2 + \frac{1}{n}\sum_{j=1}^{s}(r_j + j - \delta_{js})2^{r_j}$$

where r_j and s are defined in (5.1) and $\delta_{js} = 1$ if $j = s$ and zero otherwise.

3. Procedures $R_{t,n}$ and $R'_{t,n}$ are identical for $n = 2^r - \epsilon$, $\epsilon = 0$ or 1 and any t. The following relations are helpful in calculating some of the entries of Tables I and II:

$$(6.3) \qquad A_t(2^r) = 2^r - 1 + A_{t-1}(2^r - 1) - \sum_{j=1}^{r-1}(2^j - 1)$$

$$= r + A_{t-1}(2^r - 1),$$

$$(6.4) \qquad A_t'(2^r+1) = 2^r + \frac{1}{2^r+1}(A_{t-1}'(2^r) - 2^r + 1) + \frac{2^r}{2^r+1}(A_{t-1}'(2^r) - 2^r + r + 1)$$

$$= A_{t-1}'(2^r) + 1 + \frac{r2^r}{2^r+1} .$$

4. We can compare the results for $M_t(n)$ with a procedure (say, R_{III}) mentioned in III, Section 6 of [3] with t and n given by the sequence (4.9), which is adverse for $M_t(n)$. The results for $M_t(n)$ are given in (4.10). Under R_{III} we first find the $t = 2^r - 1$ largest (which automatically gives us the $t^{\underline{th}}$ largest and the set of size $t - 1$ containing all the items larger than the $t^{\underline{th}}$ largest). Then we order the $t - 1$ largest items say, by the Steinhaus procedure R_S. Using (2.11) of [3], (4.5) and (4.7) above for $n = t - 1 = 2^r - 2$, we obtain for $r > 2$

$$(6.5) \qquad M(R_{III}) = (r+1)(2^r-2) + \sum_{j=2}^{2^r-2} \{\log j\}$$

$$= (r+1)(2^r-2) + (2^r-2)r - 2^r + 1$$

$$= 2r(2^r-2) - 1 .$$

Using (4.10) we obtain the difference for $r > 2$

$$(6.6) \qquad M(R_{III}) - M_{2^r-1}(2^{r+1}-3) = (r-2)(2^r-2) > 0.$$

Hence $R_{t,n}$ is better than R_{III} for the sequence (4.9).

5. In [4] an inductive procedure $R_S^{(t)}$ was briefly considered for the problem of this paper, but detailed calculations were not carried out except for $t = 2$ and $t = n - 1$. For $t = 2$ it coincided with Picard's solution (see R_P and the references in [9]) and for $t = n - 1$ it coincides with the Steinhaus procedure R_S. This procedure $R_S^{(t)}$ is both E-optimal and M-optimal in a restricted class of "one-step inductive" procedures. $M(n|R_S^{(n-1)})$ is given by (4.7) for $k = n$ and

$$(6.7) \qquad E(n|R_S^{(n-1)}) = \sum_{j=2}^{n} \left(\{\log j\} - \frac{2^{\{\log j\}}-j}{j}\right)$$

$$= (1 + \{\log n\})n - 2^{\{\log n\}} - \sum_{j=2}^{n} \frac{2^{\{\log j\}}}{j} ,$$

as is found in Chapter II of [4]. For all the values of n given in Tables I and II

$$(6.8) \qquad E(n|R_S^{(n-1)}) \leq A_{n-1}(n) \leq A'_{n-1}(n),$$

but

$$(6.9) \qquad A_2(n) \leq A'_2(n) \leq E(n|R_S^{(2)}) = n - 1 + 2\sum_{j=3}^{n} \frac{1}{j} .$$

Table I: Values of $A_t(n) \equiv E\{R_{t,n}\}$.

n \ t	1	2	3	4	5	6	7	8
2	1							
3	2	$2\frac{2}{3}$						
4	3	4	$4\frac{2}{3}$					
5	4	$5\frac{2}{5}$	$6\frac{1}{2}$	$7\frac{1}{6}$				
6	5	$6\frac{2}{3}$	$8\frac{1}{5}$	$9\frac{1}{6}$	$9\frac{5}{6}$			
7	6	$7\frac{6}{7}$	$9\frac{11}{21}$	$10\frac{20}{21}$	$12\frac{4}{105}$	$12\frac{2}{3}$		
8	7	9	$10\frac{6}{7}$	$12\frac{11}{21}$	$13\frac{20}{21}$	$15\frac{4}{105}$	$15\frac{2}{3}$	
9	8	$10\frac{2}{9}$	$12\frac{1}{12}$	← not calculated →				

Table II: Values of $A'_{t,n} \equiv E\{R'_{t,n}\}$.

n \ t	1	2	3	4	5	6	7	8
2	1							
3	2	$2\frac{2}{3}$						
4	3	4	$4\frac{2}{3}$					
5	4	$5\frac{3}{5}$	$6\frac{3}{5}$	$7\frac{4}{15}$				
6	5	$6\frac{2}{3}$	$8\frac{2}{15}$	$9\frac{1}{5}$	$9\frac{13}{15}$			
7	6	$7\frac{6}{7}$	$9\frac{11}{21}$	$10\frac{20}{21}$	$12\frac{4}{105}$	$12\frac{2}{3}$		
8	7	9	$10\frac{6}{7}$	$12\frac{11}{21}$	$13\frac{20}{21}$	$15\frac{4}{105}$	$15\frac{2}{3}$	
9	8	$10\frac{2}{3}$	$12\frac{2}{3}$	$14\frac{11}{21}$	$16\frac{4}{21}$	$17\frac{13}{21}$	$18\frac{74}{105}$	$19\frac{1}{3}$

References.

[1] David, H. D. (1959). *The Method of Paired Comparisons*. Hafner Pub. Co., New York.

[2] Ford, L. R., Jr., and Johnson, S. M. (1959). A tournament problem. *Amer. Math. Monthly*, **66**, 387-389.

[3] Hadian, A. and Sobel, M. (1969). Selecting the t^{th} largest of n items using binary comparisons. *Tech. Rept. No. 121*, Dept. of Stat., Univ. of Minnesota.

[4] Hadian, A. (1969). Optimality properties of various procedures for ranking n different numbers using only binary comparisons. *Tech. Rept. No. 117*, Dept. of Stat., Univ. of Minnesota.

[5] Kislicyn, S. S. (1964). On the selection of the k^{th} element of an ordered set of pairwise comparison (Russian). *Sibirsk. Mat. Ž.*, **5**, 557-564 (MR **29** No. 2198).

[6] Moon, J. W. (1968). *Topics on Tournaments*. Holt, Rinehart and Winston, New York (see p. 48).

[7] Schreier, J. (1932). On tournament elimination system (Polish). *Mathesis Polska*, **7**, 154-160.

[8] Slupecki, J. (1949-51). On the system S of tournaments. *Colloq. Math.* **II**, 286-290.

[9] Sobel, M. (1968). On the ordering of the t best of n items using binary comparisons. *Tech. Rept. No. 113*, Dept. of Stat., Univ. of Minnesota.

[10] Steinhaus, H. (1950). *Mathematical Snapshots*. Oxford Univ. Press, New York.

[11] Steinhaus, H. (1958). *One Hundred Problems in Elementary Mathematics*. (See Problems 52 and 85 in the 1963 edition, Pergamon Press, London).

[12] Steinhaus, H. (1960). Mathematical Snapshots (New edition, revised and enlarged). Oxford Univ. Press, New York.